Energy & Buildings 194 (2019) 328–341

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enbuild

Recurrent inception convolution neural network for multi short-term load forecasting

Junhong Kim[a], Jihoon Moon[b], Eenjun Hwang[b], Pilsung Kang[a,*]

[a] School of Industrial Management Engineering, Korea University, Seoul 136-701, Republic of Korea
[b] School of Electrical Engineering, Korea University, Seoul 136-701, Republic of Korea

ARTICLE INFO

Article history:
Received 13 March 2019
Revised 1 April 2019
Accepted 19 April 2019
Available online 24 April 2019

Keywords:
Recurrent inception convolution neural network
Deep learning
Recurrent neural network
Convolution neural network
Load forecasting

ABSTRACT

Smart grid and microgrid technology based on energy storage systems (ESS) and renewable energy are attracting significant attention in addressing the challenges associated with climate change and energy crises. In particular, building an accurate short-term load forecasting (STLF) model for energy management systems (EMS) is a key factor in the successful formulation of an appropriate energy management strategy. Recent recurrent neural network (RNN)-based models have demonstrated favorable performance in electric load forecasting. However, when forecasting electric load at a specific time, existing RNN-based forecasting models neither use a predicted future hidden state vector nor the fully available past information. Therefore, once a hidden state vector has been incorrectly generated at a specific prediction time, it cannot be corrected for enhanced forecasting of the following prediction times. To address these problems, we propose a recurrent inception convolution neural network (RICNN) that combines RNN and 1-dimensional CNN (1-D CNN). We use the 1-D convolution inception module to calibrate the prediction time and the hidden state vector values calculated from nearby time steps. By doing so, the inception module generates an optimized network via the prediction time generated in the RNN and the nearby hidden state vectors. The proposed RICNN model has been verified in terms of the power usage data of three large distribution complexes in South Korea. Experimental results demonstrate that the RICNN model outperforms the benchmarked multi-layer perception, RNN, and 1-D CNN in daily electric load forecasting (48-time steps with an interval of 30 min).

1. Introduction

As the Kyoto Protocol expires and the Paris Agreement comes into effect in 2020, many countries have an obligation to ensure the reduction of carbon dioxide (CO_2) emissions and global warming [1,2]. Considerable amounts of carbon dioxide are generated during energy production (i.e., electricity, gas, etc.) through fossil fuels such as coal and oil [3,4]. South Korea is the 8th largest energy consumer in the world [5]. The South Korean government recently formulated the 'Green Building Construction Assistance Act' to serve as a legal framework for promoting eco-friendly buildings by 2020 with the goal of reducing greenhouse gas emissions by 26.9% [6,7]. The key elements of this Act include the enhancement of energy efficiency for existing and new buildings and the promotion of low-energy buildings. Electricity consumption has been steadily increasing at an annual rate of 5.4% between 2000 and 2013 owing to the expansion of facilities, increased production of semiconductors/petrochemical products/electric power plants, progressive tariffs for residential electricity consumption, and temperature effects [2]. Energy-intensive buildings are buildings that consume more than 2000 tons of oil a year such as hospitals, hotels,

Abbreviations: ANN, Artificial Neural Network; ARIMA, Autoregressive Integrated Moving Average; CNN, Convolution Neural Network; CRBM, Conditional Restricted Boltzmann Machine; DL, Deep Learning; DNN, Deep Neural Network; DT, Decision Tree; ELM, Extreme Learning Machine; EMS, Energy Management System; ESS, Energy Storage System; FCRBM, Factored Conditional Restricted Boltzmann Machine; GA-ANFIS, Genetic Algorithm—Adaptive Network-based Fuzzy Inference System; GRNN, Generalized Regression Neural Network; HVAC, Heating, Ventilation, and Air Conditioning; ICT, Information & Communication Technology; IoT, Internet of Things; IPSO-ANN, Improved Particle Swarm Optimization-Artificial Neural Network; KEPCO, Korea Electric Power Corporation; KMA, Korea Meteorological Office; LSTM, Long Short-Term Memory; LTLF, Long-Term Load Forecasting; MAPE, Mean Absolute Percentage Error; MLP, Multilayer Perception; MLR, Multiple Linear Regression; MSE, Mean Square Error; MTLF, Mid-Term Load Forecasting; MWD, Multi-resolution Wavelet Decomposition; NLP, Natural Language Processing; PDRNN, Pooling-based Deep Recurrent Neural Network; PSO, Particle Swarm Optimization; PV, Photovoltaic; RBM, Restricted Boltzmann Machine; ReLU, Rectified Linear Unit; RF, Random Forest; RICNN, Recurrent Inception Convolution Neural Network; RMSE, Root Mean Square Error; RNN, Recurrent Neural Network; SFOA, The Fruit Fly Optimization Algorithm with Decreasing Step Size; SNN, Shallow Neural Network; SRWNN, Self-Recurrent Wavelet Neural Network; STLF, Short-Term Load Forecasting; SVR, Support Vector Regression; S2S, Sequence to Sequence; VSTLF, Very Short-Term Load Forecasting; WNN, Wavelet Neural Network; 1-D CNN, 1-Dimensional Convolution Neural Network.

* Corresponding author.
E-mail address: pilsung_kang@korea.ac.kr (P. Kang).

https://doi.org/10.1016/j.enbuild.2019.04.034
0378-7788/

Nomenclature

BN	batch normalization
b	bias
c	Cell state
EoM	the last day of a month
F	convolution filter
f	forget gate
$f(x)$	predicted value
g	memory cell
h	hidden state
i	input gate
L	actual load
$\hat{L}$	predicted load
n	number of observations
O	value of convolution layer
o	output gate
ReLU	Rectified Linear Unit
tanh	hyperbolic tangent
W	weights
X	input variable set
x	input vector
y	actual value

Subscripts

f	forget gate
g	memory cell
i	input gate
i	the ith value in a particular layer
l	layer index of the convolution layer
m	the number of convolution operation
o	output gate
t	time sequence index

educational facilities, commercial buildings, telecommunications facilities, and apartment buildings [7]. For these heavy energy consumption buildings, smart grid systems combined with renewable energy or ESS are mandatory for efficient management of total energy consumption [8].

A smart grid is an innovative power system platform that exchanges real-time power information between suppliers and consumers through various ICT technologies such as wired/wireless communication, control, and sensors [5]. In particular, EMS, which is one of the key elements of the smart grid, collects and analyzes data related to energy consumption, such as cooling device usage and lighting usage, within the smart grid system and determines ways of saving energy on the demand side [9]. On the supply side, EMS makes predictions about the amount of energy to be used in the future, thereby effectively scheduling power generation and ESS based on forecasts to limit energy consumption and storage costs [10,11]. Owing to these reasons, new renewable energy and ESS markets have been rapidly growing. To effectively operate an ESS-based smart grid, accurate electric load forecasting is necessary [12]. Accurate electric load forecasting provides economic benefits by storing energy at night when electric charges are relatively low and emitting electricity during the day when electric charges are high. Furthermore, if the smart grid market proliferates, electric power producers will be able to efficiently produce electricity by adaptively adjusting the priorities of high cost or less climate-friendly power generation methods such as old power plants or fossil fuel-based power plants [13].

Electric load forecasting can be divided into four main categories depending on the range of forecasts [14]. When the prediction is made within a day or even shorter periods, it is called VSTLF. When the prediction is made in the range of one day to two weeks, it is called STLF. MTLF forecasts ranging from two weeks to three years and forecasts over three years are called LTLF. Generally, VSTLF with a forecast cycle shorter than an hour plays a significant role in scheduling smart grid operations associated with ESS and renewable energy. In addition, it can contribute to economic daily scheduling, including peak response by aggregating the short period forecasts to predict a total daily load.

Electric load forecasting is not an easy task because building's energy consumption patterns are complex and uncertain external factors can cause a shift in the demand curve [15]. Factors affecting fluctuations in electric load include low-level systems composed of an architectural structure, thermal properties of physical materials, time zones, electricity rates, special events, resident schedules, climatic conditions, and lighting or HVAC [16]. In addition, when forecasting electric load, the highly complex correlation associated with energy consumption between the current time and the previous time should be effectively considered. To forecast the exact electric load, three methods have been primarily studied: conventional statistical techniques, machine learning using MLP and SVR, and deep learning. [17]. There have been several recent studies on improving short-term load forecasting performance using LSTM-based RNNs that can effectively process sequential data [18–21]. However, these studies have some limitations in that the predicted future hidden state vector is not used when forecasting a specific electric load using LSTM. In addition, because these studies do not use all past hidden state vectors but only use one preceding hidden state, it is impossible to correct when a hidden state vector is incorrectly generated at the prediction time. To address these problems, we propose a novel multi short-term load forecasting model based on RICNN that composed of RNN and 1-D CNN of an inception module. To verify the effectiveness of the proposed model, we perform electric load forecasting for the next 24 h with a time interval of 30 min for three distribution industrial complexes with high electric load in South Korea. The forecasting performance of the proposed RICNN is compared with MLP and other deep learning-based load forecasting methods: 1-D CNN and RNN.

The rest of this paper is organized as follows. In Section 2, we briefly review related works on STLF and VSTLF. In Section 3, we describe the data preprocessing procedures that transform the historical load consumption data and external information to RICNN input vectors. Section 4 demonstrates the multi short-term load forecasting model based on deep learning algorithms, including the proposed RICNN. In Section 5, we describe the experimental design, and the experimental results are presented in Section 6. In Section 7, we conclude this paper with some future research directions.

2. Related Work

In this section, we introduce studies on STLF (including VSTLF) that focus on the efficient operation of smart grid systems. Prior studies on conventional statistical techniques and machine learning for forecasting electric load [14,22–24], and recent studies based on DNN [25] have further improved prediction performance. Table 1 summarizes STLF-related studies based on statistical techniques and machine learning.

The statistical approach exhibits the advantage of being able to easily determine the correlation or causal relationship between the independent variables and the dependent variable. Vaghefi et al. [26] proposed a Cochrane–Orcutt estimation technique that combines MLR and a seasonal autoregressive moving average model that effectively forecasts cooling and electric load. This model adaptively assumes the advantages of both time series and regression methods so that it can update forecast values each time that new information on cooling and electricity load is received. Grzegorz [27] developed a forecasting model based on stepwise

Table 1
Summary of short-term load forecasting based on statistical techniques and machine learning.

Authors (Year)	Target	Input Variable	Forecasting Method	Highlight
Vaghefi et al. [26]	A CCHP plant at University of California, Irvine	Historical load, weather information, time information	Cochrane-Orcutt estimation	Utilizing the advantage of both time series and regression methods
Dudek [27]	Polish power system	Historical load	Principal components regression, partial least-squares regression	Proposed univariate models based on linear regression and patterns of daily cycles of load time series
Fard and Akbari-Zadeh [28]	Fars Electric Power Company in Iran	Historical load	ARIMA, ANN, DWT	Applying several ANNs to the details and approximation components of the residuals signal.
Sudheer and Suseelatha [29]	California and Spain energy markets	Historical load	Wavelet transform, triple exponential smoothing, weighted nearest neighbor	Capturing the trend and seasonality factors of the deterministic component of load data.
Ke et al. [30]	The Centennial Campus at the North Carolina State University	Historical load, weather information, time information	Polynomial regression, similar day approach, MLR	MLR showed the advantage for forecasting the trend of load variations.
Gerossier et al. [31]	A neighborhood comprising 226 individual buildings in Évora, Portugal	Historical load, weather information	Quantile smoothing splines regression	Providing probabilistic forecasts by computing a list of quantiles.
Chen and Tan [32]	The mall and the hotel buildings	Historical load, weather information	Wavelet decomposition, SVR	Applying a hybrid SVR model combined with MWD, which represent stationary and nonstationary sequences respectively.
Jain et al. [33]	A multi-family residential building located on the Columbia University campus in New York	Historical load, weather information, time information	SVR	Broadened their exploration to examine the impact spatial and temporal granularity had on sensor-based forecasting models
Amber et al. [34]	An Administration building located at the Southwark campus of London South Bank University in London.	Weather information, Time information	Genetic programming, MLR	GP model achieved slightly better forecasts, but the training time consumed by the GP model was more than the MR model.
Grolinger et al. [35]	A large event-organizing venue, located in Ontario, Canada	Historical load, time information, event information	ANN, SVR	Daily data intervals resulted in higher consumption prediction accuracy than hourly or 15-min readings.
Jurado et al. [36]	Three buildings of the UPC (Universitat Politècnica de Catalunya)	Historical load, time information	RF, ANN, FIR, ARIMA	The approaches discussed generate fast and reliable models, with low computational costs.
Moon et al. [37]	A private university in Seoul, South Korea	Historical load, Weather information, Time information	RF	Considering the electric load pattern by using the moving average method.

regression and lasso regression analysis, which reflects the expected output patterns and electric load cycle patterns of a day. Using principal component regression or partial least-squares regression, the number of predictors can be reduced to only one, which enables the visualization of the regression function. These models yield high accuracies compared to ARIMA, exponential smoothing, ANN, Nadaraya–Watson estimator, and so on. Fard and Akbari-Zadeh [28] proposed a novel hybrid forecasting method based on discrete wavelet transform (DWT), ARIMA, and ANN. By using the DWT, the residual signals of the ARIMA model were decomposed into detailed and approximated sub-parts, each of which were then modeled by an appropriate ANN. This hybrid model demonstrated notable performance (MAPE: 0.4004) higher than ARIMA (MAPE: 2.3001), AR (MAPE: 2.5008), ANN (MAPE: 1.9569), and SVR (MAPE: 1.8051). Sudheer and Suseelatha [29] proposed a hybrid forecasting model for accurate STLF based on wavelet transform, triple exponential smoothing (TES), and weighted nearest neighbor (WNN). This model demonstrates the capability of the triple exponential smoothing model in capturing the trend and seasonality factors of the deterministic component of load data. The accuracy of this hybrid model outperformed that of the Holt-Winters model, WNN, and Haar wavelet combined WNN model. Ke et al. [30] proposed a short-term load forecasting model using curve fitting, similar day method, and MLR to analyze the characteristics of the electric load of a university campus by considering the environmental factors closely related to electric load such as time of day, temperature, humidity, etc. In this experiment, MLR demonstrated the advantage of forecasting a trend of load variations and was verified to be more accurate than direct curve fitting based on polynomial regression and the similar day approach. Gerossier et al. [31] presented a forecasting model for hourly household electric load based on quantile smoothing spline regression using three input variables: the previous day's hourly load, last week's mid-load, and temperature. They computed the mean of the predicted quantile distribution and used it as a single-point forecast. These statistical approaches demonstrate good performance for simple demand patterns but, somewhat, inaccurate prediction performance for complex demand patterns [4,38]. In addition, it is difficult to assign an appropriate weight to variables in which the input variable has a nominal or nonlinear correlation using linear regression analysis [23,30]. To overcome the disadvantages associated with this statistical method, many studies have used machine learning to improve superior STLF results [23,24].

Chen and Tan [32] developed a hybrid SVR-based prediction model using MWD. This model predicted the hourly electric demand intensity at hotels and shopping malls. All error values of the hybrid model with different metrics were smaller than those of the pure SVR model. Jain et al. [33] developed a building energy forecasting model using SVR. Energy consumption data was collected from multi-family residential buildings located at the Columbia University campus in New York City. Their predictions included those related to examine the impact of temporal (daily, hourly, 10-min intervals) and spatial (the whole building, by floor, by unit) granularity. The most effective models were built with hourly consumption at the floor level. In particular, spatial granularity was demonstrated to have a substantial impact on the predictive power of sensor-based forecasting models because granular data at the floor and individual unit levels produced better predictions. Amber et al. [34] developed two models, MR and genetic programming (GP), to forecast the daily electric load of an administration building located at the Southwark campus of London South Bank University in London. Although the GP model achieved a lower total absolute error (TAE) of 6% compared to 7% in the MR model, its training time was longer than the MR model. Grolinger et al. [35] developed two prediction models based on SVR and ANN to consider the typical external factors and event information and compared power consumption predictions by day, hour, and 15-min intervals for a large entertainment building in Canada. Based on daily data, the ANN model achieved better accuracy than the SVR. However, based on hourly and 15-min data, there was no definitive dominance of one approach over another. Jurado et al. [36] predicted and compared the hourly electric load of three buildings in Barcelona using various machine learning methods such as RF, ANN, fuzzy inductive reasoning (FIR), and ARIMA. One of the aims of these experiments was to understand how the model's accuracy is affected by the insertion of new input variables without increasing computational cost. To do so, an analysis of prediction errors involving the number of most important variables, including the depth of the past values in the feature selection process (FSP), was performed. In this experiment, FIR provided better forecasts followed by RF, ANN, and ARIMA. Moon et al. [37] performed daily electric load forecasting for university campuses based on moving average and an RF-based two-stage forecasting model. They considered the electric load pattern by using a moving average method according to the day of the week. They then provided short-term load forecasting using RF and evaluated its performance based on time series cross-validation. In this experiment, RF demonstrated higher prediction accuracy than SVR and ANN.

As the performance of computers has significantly improved, various ANN-based electric load forecasting models have been developed [30]. Li et al. [39] proposed an IPSO-ANN method by combining the PSO algorithm and ANN and compared its prediction performance with that of ANN and genetic algorithm-artificial neural networks (GA-ANN) [40]. Bagnasco et al. [41] predicted the electric load of the Cellenic hospital building in Tunino using ANN. Chitsaz et al. [42] predicted the daily and peak electric load of two different buildings using SRWNN, which involves an easy parameter optimization process, and they compared MLP and WNN in terms of prediction performance. Hu et al. [43] proposed a GRNN-based STLF model using SFOA to select an optimal diffusion parameter σ that determines the performance of the GRNN. The performance of the proposed SFOA-GRNN model was verified by comparing with the performance of other ANN in terms of prediction errors. The RMSE of a BP neural network for STLF was 0.024, while the RMSE of the SFOA-GRNN model was 0.0018. . Zeng et al. [44] proposed a hybrid learning method that combines ELM and switching delayed PSO (SDPSO) algorithms for accurate STLF and reported superior performance than existing ELM. Reddy [45] proposed a back-propagation approach based on the Bat algorithm for STLF, reflecting weather factors such as temperature and humidity. The Bat algorithm-based back propagation approach was determined to be an efficient prediction method because it significantly reduced learning time compared with the vanilla ANN-based approach. Mordjaoui et al. [46] predicted daily electric load using a dynamic neural network. The proposed technique demonstrated that the accuracy and efficiency of prediction can be enhanced by simulating power data collected from the French Transmission System Operator website.

DNN has been recently employed in various research fields such as electric load forecasting because of advancements in IoT and hardware [20,21,48]. Ryu et al. [47] developed and compared two DNN-based electric load forecasting models consisting of RBM and ReLU without pre-training. They confirmed that models using ReLU were easier to learn and performed better than SNN. Kuo and Huang [48] proposed a novel electric load forecasting model based on 1D-CNN and a pooling layer. They suggested a new research direction in electric load forecasting through the comparison of traditional machine learning methodology and MLP. Zhang et al. [17] constructed a short-term load forecasting scheme based on RNN by considered multiple time series (MTS) consisting of four information sequences (short-term, cycle, long short-term, and cross-long short-term). Abdel-Nasser and Mahmoud [18] developed five LSTM-based PV output forecasting models to compare

and evaluate prediction accuracy performance. They concluded that model3 (LSTM for regression with time steps) demonstrated the best performance compared to model1 (basic LSTM network for regression), model2 (LSTM for regression using the window technique), model4 (LSTM with memory between batches), and model5 (stacked LSTMs with memory between batches). Marino et al. [19] developed an S2S-based LSTM forecasting model for building energy consumption forecasting. In their study, the S2S-based LSTM architecture yielded improved performance over basic LSTM architecture in both 1-h and 1-min increments. Mocanu et al. [16] proposed five methods of predicting the energy consumption of residential buildings at various time periods. The proposed method predicted electric load based on accurate FCRBM compared to deep learning-based CRBM. FCRBM and CRBM exhibited greater robust prediction performance than benchmark algorithms as prediction horizon increases. Shi et al. [20] proposed a pooling scheme and a PDRNN based on RNN for home short-term electric load forecasting. The pooling technique is based on past historical data and time information of neighbors demonstrating similar energy consumption patterns in households to be tested. An LSTM-based RNN was then used to build a forecasting model. Therefore, the proposed PDRNN learned not only individual load patterns but also common load characteristics and uncertainties, demonstrating excellent prediction performance for all households. Kong et al. [21] used density-based clustering to evaluate and compare mismatches between the load on system levels and the load in individual households to justify the use of LSTM. They then performed short-term load forecasting of individual households based on LSTM using past energy consumption, sequence of the time of day, day of the week, and binary holiday to recognize the lifestyle patterns of residents. In summary, the LSTM-based forecasting model yielded much better performance for single-meter load forecasting than the BPNN-based forecasting model because the advantage of the LSTM is its ability to track and learn the temporal relationships of energy consumption.

In summary, the LSTM-based RNN has demonstrated improved performance in short-term load forecasting compared with traditional statistical or machine learning-based models. However, the LSTM-based RNN exhibits two significant limitations: (1) it does not use the hidden state vector of the future and (2) it does not fully use the past hidden state vectors except for immediately preceding time. In other words, load forecasting is solely based on the current LSTM hidden state vector. Therefore, there is no opportunity to correct inaccurate hidden state vectors at the predicted time. To address this problem, we propose a novel multi short-term load forecasting model based on RICNN and composed of RNN and 1-D CNN with an Inception module [49,50].

3. Data collection and preprocessing

In this paper, we used actual sensor data collected from smart meters at three large distribution complexes in South Korea.

Table 2
Raw data information from each area.

Area	Raw data collection period (day)	Land area	Building floor area
Incheon	2013-02-01–2014-06-19 (503)	235,226 m^2	265,769 m^2
Gwangju	2013-01-18–2014-06-19 (517)	126,219 m^2	77,685 m^2
Shihwa	2013-01-05–2014-06-19 (530)	107,810 m^2	186,533 m^2

Table 2 presents the data collection period and the area information of the distribution complexes.

The term "sensor data" used in this paper refers to the data accumulated in demand at a certain time interval. The time unit of collected data is 30 min. The data recorded in one sensor denotes power consumption from one store. We first integrated all the electric loads of the distribution complex at the same time zone and used them as the target variables for the forecasting model.

To build an accurate electric load forecasting model, it is necessary to define an appropriate set of input variables. Table 3 presents the input variables and the target variable that we considered, which is described in more detail in the next chapter.

3.1. Sequence information

Because time series data indicates a trend in electric load, we consider all the variables that can express temporal data such as month, day, hour, minute, day of the week, and holiday. Because the month, day, hour, and minute data exhibit periodic properties, they should not be represented by sequential values. For example, although 23 o'clock and 0 o'clock are adjacent, their difference is 23 in the sequence format. To reflect the periodicity of temporal data, we used the Eqs. (1)–(7), which eventually enhances sequence data in the one-dimensional space to continuous data in the two-dimensional space [51]. In the case of minutes, there are only two cases (0, 30). Therefore, the hour and minutes data can be reflected in the corresponding time as shown in Eq. (1) and then applied to Eqs. (2) and (3). EoM_{month} denotes the last day of the month. For example, $EoM_{January}$ is 31, $EoM_{February}$ is 28/29, EoM_{March} is 31, etc. Therefore, Eqs. (2)–(7) are used as six input variables to represent the date and time of the prediction time point.

$$hourmin = \text{hour} + \left({}^{minute}/_{60}\right) \tag{1}$$

$$hour_x = \sin\left(\left({}^{360}/_{24}\right) \times hourmin\right) \tag{2}$$

$$hour_y = \cos\left(\left({}^{360}/_{24}\right) \times hourmin\right) \tag{3}$$

$$day_x = \sin\left(\left({}^{360}/_{EoM_{month}}\right) \times day\right) \tag{4}$$

$$day_y = \cos\left(\left({}^{360}/_{EoM_{month}}\right) \times day\right) \tag{5}$$

Table 3
Variable description from raw data (strings in parentheses are the data type).

No	Variable name	Description	No	Variable name	Description
1	Hour_x	Sine value at the hour (numeric)	12	Friday	Friday (binary)
2	Hour_y	Cosine value at the hour (numeric)	13	Saturday	Saturday (binary)
3	Day_x	Sine value at the day (numeric)	14	Sunday	Sunday (binary)
4	Day_y	Cosine value at the day (numeric)	15	Temperature	Adjusted temperature (numeric)
5	Month_x	Sine value at the month (numeric)	16	Humidity	Humidity (numeric)
6	Month_y	Cosine value at the month (numeric)	17	Wind speed	Wind speed (numeric)
7	Holiday_today	Weekdays/holidays status (binary)	18	Off-peak	Off-peak status (binary)
8	Monday	Monday (binary)	19	Mid-peak	Mid-peak status (binary)
9	Tuesday	Tuesday (binary)	20	On-peak	On-peak status (binary)
10	Wednesday	Wednesday (binary)	21	Sensor	Number of smart meter sensors (numeric)
11	Thursday	Thursday (binary)	22	Electric Load (Target variable)	30-min demand load (numeric)

$$month_x = \sin\left(\left({}^{360}/_{12}\right) \times month\right) \quad (6)$$

$$month_y = \cos\left(\left({}^{360}/_{12}\right) \times month\right) \quad (7)$$

Most industrial distribution complexes operate on weekdays and do not operate on public holidays such as Saturdays, Sundays, and national holidays. Therefore, weekdays and holidays generally exhibit different electric load patterns. To reflect these factors, the day of the week and holiday data, which are the nominal measures, are reflected in the prediction model. To represent a nominal scale, we defined a vector of 0 or 1 for each distribution complexes depending on the 8-dimensional feature vector composed of the seven days of the week and holidays. Finally, we used 14 information sequences that denote the prediction time point.

3.2. Weather information

The use of products with high-power consumption, such as air conditioners and radiators, is closely related to weather conditions such as temperature, humidity, wind speed, solar radiation, and cloud cover [34,39,40,44,45,47]. Therefore, input variables derived from weather information are commonly used for short-term load forecasting in many studies [14,22–24]. KMA's Digital Forecast provides date, weather, temperature, wind speed, and humidity predictions for every region unit in South Korea as shown in Fig. 1. However, it does not provide forecast values for solar radiation and cloud cover. Therefore, we use three meteorological parameters that are composed of temperature, wind speed, and humidity, which demonstrate great influence on electric load based on local forecasts.

We used the original values of humidity and wind speed provided by the KMA, but the temperature was adjusted to more precisely reflect the relationship between seasonal factors and electric power consumption. In countries with four distinct seasons such as South Korea, electric loads are relatively higher during the summer and winter than during the spring and autumn because air conditioners and heating appliances are heavily used in the summer and winter, respectively. This indicates that electric load and temperature are not linearly correlated: electric load is high when the temperature is very high or very low. Therefore, we adjusted the temperature as the absolute difference in the annual average temperature, i.e., 12.5 °C, as shown in Eq. (8):

$$Adjusted_{Temp} = |12.5 - \text{Temperature}| \quad (8)$$

Table 4 presents the Pearson correlation coefficient between the adjusted temperature and the electric load. Although the correlation coefficient of the electric load to the original temperature was negative, Pearson correlation coefficient was reversed when the adjusted temperature was used.

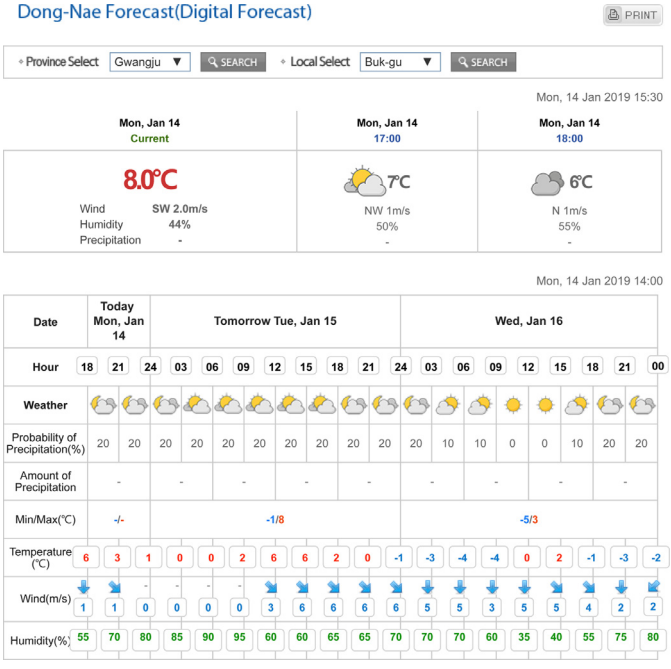


Fig. 1. Example of digital forecast by KMA.

Table 4
Comparison of the Pearson correlation coefficient.

Area	Original correlation coefficient	Adjusted correlation coefficient
Incheon	−0.076	0.147
Shihwa	−0.226	0.153
Gwangju	−0.007	0.119

Table 5
Seasonal load time zone in Korea.

Time zone category	March–October	November–February
Off-peak	23:00–09:00	23:00–09:00
	09:00–10:00	09:00–10:00
Mid-peak	12:00–13:00	12:00–17:00
	17:00–23:00	20:00–22:00
On-peak	10:00–12:00	10:00–12:00
	13:00–17:00	17:00–20:00
		22:00–23:00

3.3. Electricity rates

Because one of the operational goals of the smart grid is to reduce the cost of electricity [1,52], many electric load forecasting studies have considered electricity rates as one of the input variables [23,24]. Therefore, we also considered information on electricity rates as input variables. KEPCO, which provides electricity to the South Korean market, charges different electricity rates according to the degree of power demand (i.e., off-peak, mid-peak, and on-peak), the purpose of the building (residential or industrial), and the season or the month. Table 5 presents electricity rates according to the time zone in the industrial area in which the distribution industrial complex is located. Similar to the day and holiday information described previously, we developed three-dimensional one-hot vectors that indicate a time zone category.

3.4. Number of sensors

The target variable of our forecasting model is the building-level electric load, which is the integrated electric load determined by the smart meters installed at individual stores. Because the number of running smart meters is highly correlated with total energy consumption in the building, it should be considered as one of the input variables in the forecasting model. For example, if a new store opens, the number of smart meters is increased by one and the electric load is increased because the new store uses some amount of electricity. If an existing store shuts down, on the other hand, the number of smart meters decreases by one and the electric load consequently decreases. The number of running smart meters can be obtained by opening and closing the schedule of individual stores in the building.

3.5. Historical load data

In addition to the 21 input variables described above, we used recent historical electric loads as another type of input variable. We considered 30-min power demands in the past three days, five days, and seven days. In particular, if the historical electric loads for the past three days are used, 144 input variables (2 measurements/h × 24 h/day × 3 days) are generated, based on which the following 48 electric loads (2 measurements/h × 24 h/day) are predicted.

4. Multi short-term load forecasting based on deep neural networks

Because the proposed RICNN is based on RNN and 1-D convolution, we first introduce the benchmark forecasting models based on MLP, 1-D CNN, and RNN, prior to demonstrating the RICNN model.

4.1. Multilayer perceptron

MLP is a feed forward neural network architecture comprising three layers: the input layer, hidden layer, and output layer. The most fundamental MLP structure has only one hidden layer, but MLP with more than two hidden layers has been consistently adopted in many real-world applications because of its high predictive power [9,15,47]. The MLP used in this paper includes five hidden layers as shown in Fig. 2. X refers to the input variables that are not related to historical electric loads, whereas L refers to the input variables related to historical electric loads. The primary difference between X and L is that the former can be obtained prior to forecasting, but the latter cannot be obtained. When we make a prediction, we also used the future X variables. For example, we use information to determine when a holiday is approaching to predict the electric loads of the day. L and $\hat{L}$ denote the actual and predicted loads, respectively. Two different MLP structures are used in this study as shown in Table 6. MLP-B has more hidden units in each dense layer than MLP-A.

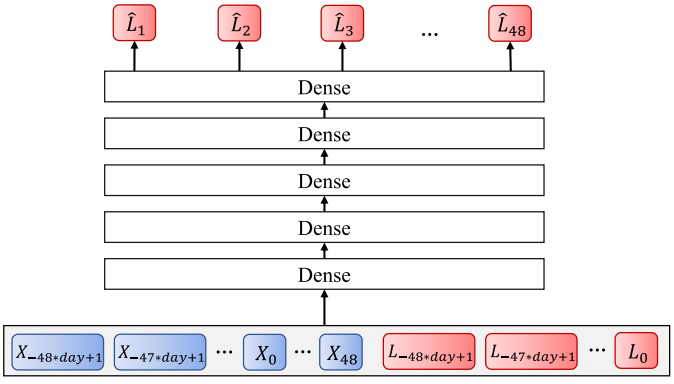


Fig. 2. Architecture of multilayer perceptron for load forecasting.

Table 6
Multilayer perceptron parameter details.

MLP-A	MLP-B
Dense-128	Dense-256
Dense-128	Dense-256
Dense-128	Dense-256
Dense-128	Dense-256
Dense-128	Dense-256
Dense-48	Dense-48

4.2. 1-Dimensional convolution neural network

CNN has evolved in the field of computer vision with well-known structures such as AlexNet, VGGNet, Inception, ResNet, and DenseNet [53–57]. Recently, CNN demonstrated better performance than other machine learning algorithms such as MLP and SVM not only in the NLP domain [58] but also in electricity load forecasting [47,59-60]. In this study, 1-D CNN is used to cope with the time series characteristics of electric load forecasting. An example of the operation of a 1-D CNN is presented in Fig. 3. In Eq. (9), O_i^l refers to the ith value in the lth layer. When the feature map of the $l-1$ layer is of width W, height 1, and channel C, it is denoted as $O^{l-1} \in \mathbb{R}^{W\times\ 1\ \times C}$ and the parameter $F^l \in \mathbb{R}^{k\times\ 1\ \times C}$ in the lth layer is a weight that is learned by the gradient descent method and serves to identify a local pattern. $*$ is a convolution product with inverted weights, while *BN* and *ReLU* denote the batch normalization and rectified linear unit activation function.

$$O_i^l = ReLU\left(BN\left(O^{l-1} * F^l\right)_i\right) = ReLU\left(BN\left(\sum_m O_{i+m}^{l-1} * F_m^l\right)\right) \quad (9)$$

The structure of the 1-D CNN used in this study is illustrated in Fig. 4. The X variables and L variables available from the past to the present are used as the first input set, and the second input set consists of the X variables for the following 48 time points with 30 min intervals. Two 1-D CNN structures are used in this study as shown in Table 7.

4.3. Recurrent neural network

RNN is one of the neural network architectures designed to process sequential data and has demonstrated superior performance in time-series data such as voice and text [61–65]. LSTM, proposed by Hochreiter and Schmidhuber [66], is the most commonly used RNN variant that can learn long-term dependency. In this study, we used the basic LSTM cell because of its structural simplicity

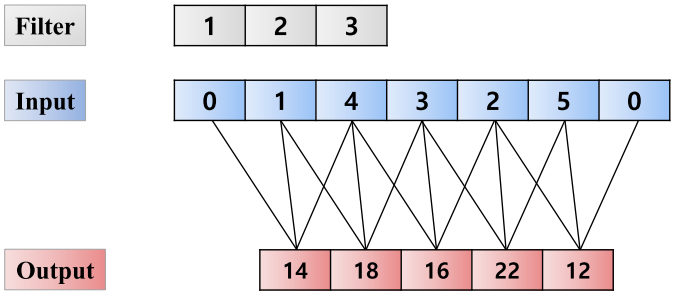


Fig. 3. Example of one-dimensional convolution (stride = 1, zero padding = 1).

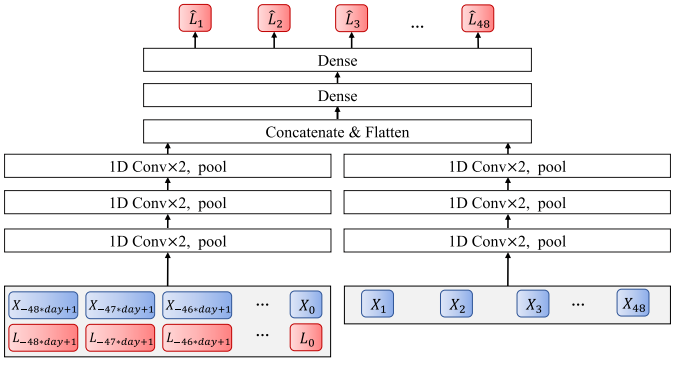


Fig. 4. Architecture of 1-D convolution neural network for load forecasting.

Table 7
Network parameters for two 1-D CNN models.

1-D CNN-A			1-D CNN-B		
1-D Conv3-32		1-D Conv3-32	1-D Conv3-64		1-D Conv3-64
1-D Conv3-32		1-D Conv3-32	1-D Conv3-64		1-D Conv3-64
	Maxpool			Maxpool	
1-D Conv3-64		1-D Conv3-64	1-D Conv3-128		1-D Conv3-128
1-D Conv3-64		1-D Conv3-64	1-D Conv3-128		1-D Conv3-128
	Maxpool			Maxpool	
1-D Conv3-128		1-D Conv3-128	1-D Conv3-256		1-D Conv3-256
1-D Conv3-128		1-D Conv3-128	1-D Conv3-256		1-D Conv3-256
	Maxpool			Maxpool	
	Concatenate			Concatenate	
	Dense-128			Dense-256	
	Dense-128			Dense-256	
	Dense-48			Dense-48	

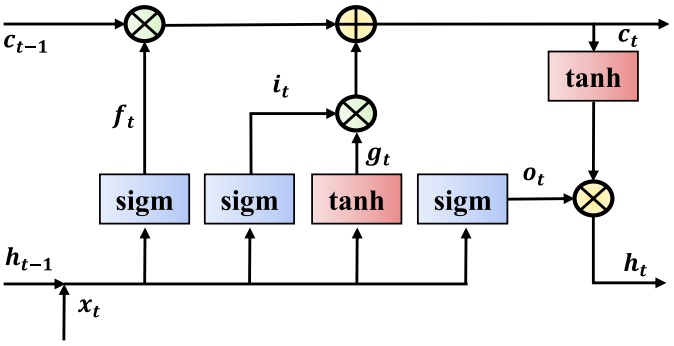


Fig. 5. Architecture of LSTM cell.

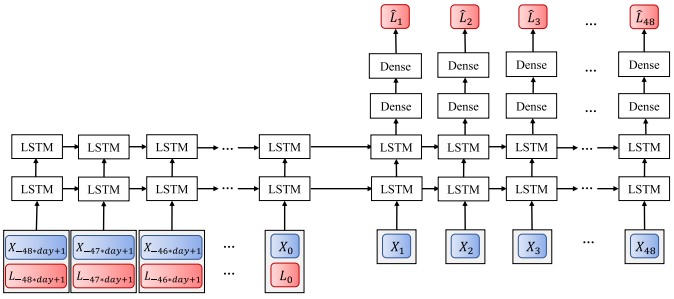


Fig. 6. Architecture of recurrent neural network for load forecasting.

and performance indifference to other sophisticated RNN variants [67,68]. Because power load exhibits an obvious long/short-term time series pattern, LSTM has attempted to improve electricity load forecasting [17–21].

The LSTM cell structure is illustrated in Fig. 5 and the operating equations are provided in Eqs. (10)–(15). In Fig. 5, the line at the top is the cell state c, which refers to the internal memory. The line across the bottom is the hidden state, and the *i, f, o*, and *g* gates are designed to resolve the vanishing gradient problem [65]. While training, each gate learns the weight and bias, respectively. The forget gate f_t regulates the amount that can pass though the previous hidden state h_{t-1}. The input gate i_t defines the computed value from the current input x_t. The output gate o_t regulates the amount of the hidden state h_{t-1} in the next sequence. The internal hidden state g_t is computed based on the input x_t and the previous hidden state h_{t-1}.

$$f_t = \sigma\left(W_f * [h_{t-1}, x_t] + b_f\right) \tag{10}$$

$$i_t = \sigma\left(W_i * [h_{t-1}, x_t] + b_i\right) \tag{11}$$

$$o_t = \sigma\left(W_o * [h_{t-1}, x_t] + b_o\right) \tag{12}$$

$$g_t = \tan h(W_g * [h_{t-1}, x_t] + b_g) \tag{13}$$

$$c_t = (c_{t-1} \otimes f_t) \oplus (g_t \otimes i_t) \tag{14}$$

$$h_t = \tanh(c_t) \otimes o_t \tag{15}$$

The RNN architecture used in this study is shown in Fig. 6, and Table 8 presents the parameters of the two different RNN models: RNN-B comprises 256 hidden nodes, which is twice the number of RNN-A.

Table 8
Model parameters of RNN.

RNN-A	RNN-B
LSTM-128	LSTM-256
LSTM-128	LSTM-256
Dense-128	Dense-256
Dense-128	Dense-256
Dense-1	Dense-1

4.4. Recurrent inception convolution neural network

There have been many recent attempts to combine CNN and RNN in various fields such as computer vision [69,70], NLP [71], and sound [72,73]. These attempts yielded better performance than single CNN-based or RNN-based models because of the synergy effect of CNN (ability to capture local significant relationships) and RNN (ability to handle a variable length of sequential data). In this study, we proposed an RICNN model for electric load forecasting as shown in Fig. 7 based on the inception module, which was first proposed by GoogLeNet [55]. The RNN model shown in Fig. 6 does not use the future hidden state vectors for forecasting. In addition, this architecture does not directly use the past hidden states except for the last hidden state. Therefore, the forecast is made solely based on the last hidden state vector. In other words, there is no opportunity to calibrate when unfavorable previous hidden state vectors are generated. To address this problem, we included an inception module with four different sizes of 1-D convolution between the last LSTM layer and the first fully connected layer. By adding the inception module, RICNN can make predictions based not only on past information but also on predicted future information.

In the inception module in the RICNN model, the hidden state vectors of each time step are concatenated to construct a hidden state matrix. If the 1-D convolution filter size is one in the incep-

Table 9
Recurrent inception convolution neural network parameter details.

	RICNN-A				RICNN-B		
	LSTM-128 LSTM-128 Concatenate				LSTM-256 LSTM-256 Concatenate		
1-D Conv 1-32	1-D Conv 3-32	1-D Conv 5-32	1-D Conv 7-32	1-D Conv 1-64	1-D Conv 3-64	1-D Conv 5-64	1-D Conv 7-64
	Concatenate Dense-128 Dense-128 Dense-1				Concatenate Dense-256 Dense-256 Dense-1		

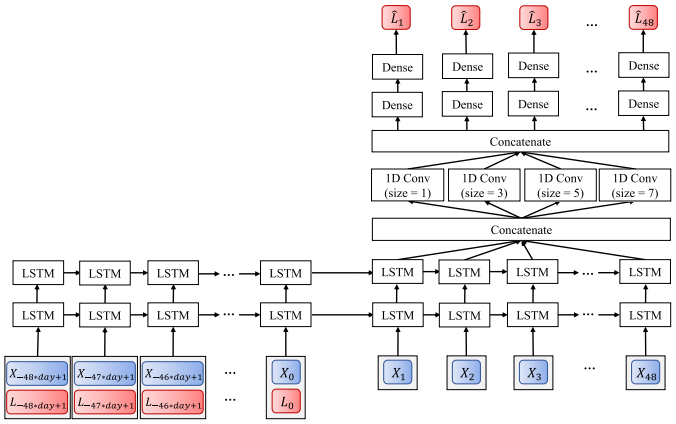


Fig. 7. Architecture of recurrent inception convolution neural network for load forecasting.

tion module, it is equal to an RNN structure that uses the current hidden state vector only. A filter size of three refers to the prediction of electricity load by considering the preceding and the following hidden state vector, i.e., it considers the information available 30 min before and after the current time step. When the filter size is five, RICNN assumes the two preceding and two following hidden vectors when predicting the electricity load, i.e., it considers the information available 60 min before and after the current time step. The inception module with a filter size of seven can be understood in a similar way. In doing so, we can expect that an optimized set of features are generated from the current and nearby past and future hidden state vectors. Therefore, although an incorrect hidden state vector is generated at the current time, the 1-D inception module can correct it and send it to the dense layer. After passing through the 1-D inception module, the hidden state vectors from each filter are concatenated and assigned to the dense layer for each sequence. Table 9 presents the network hyper parameters used in two RICNN models: RICNN-B has a more complex structure than RICNN-A.

5. Experiment design

5.1. Data partition

Raw data is collected from three local industrial distribution complexes. We divided the entire dataset into a training dataset (75%) and a test dataset (25%). The training and the test periods for the three local complexes are presented in Table 10.

5.2. Experiment design

For all forecasting models, batch normalization [74] was used after all dense and convolution layers, and ReLU [53] was used as the activation function. In addition, we used the Xavier initialization [75] for weight initialization. We set the batch size to 256 and

Table 10
Data partition for each cluster area.

Area	Training set period (day)	Test set period (day)
Shihwa	2013-01-05–2014-02-06 (398)	2014-02-07–2014-06-19 (132)
Incheon	2013-02-01–2014-02-13 (377)	2014-02-14–2014-06-19 (126)
Gwangju	2013-01-18–2014-02-09 (388)	2014-02-10–2014-06-19 (129)

Table 11
Input variable format for Multilayer perceptron.

Amount of past days to use for training	Training data size
3 days	(4176)
5 days	(6288)
7 days	(8400)

Table 12
Input variable format for 1-D CNN.

Amount of past days to use for training	Training data size 1 (Past–now)	Training data size 2 (Future)
3 days	(22,144,1)	(21,48,1)
5 days	(22,240,1)	(21,48,1)
7 days	(22,336,1)	(21,48,1)

Table 13
Input variable format for RNN and RICNN.

Amount of past days to use for training	Training data size 1 (Past–now)	Training data size 2 (Future)
3 days	(22,144)	(21,48)
5 days	(22,240)	(21,48)
7 days	(22,336)	(21,48)

used RMSProp [76] for the optimization method. The learning rate began at 10^{-3} and decayed by 10^{-1} for every 5000 iterations until 15,000 iterations were completed. The Huber loss [77] with $\delta = 1$ was used because it is less sensitive to outliers than the MSE loss function. The Huber loss equation is shown in Eq. (16):

$$L_\delta(y, f(x)) \begin{cases} \frac{1}{2}(y - f(x))^2 & for\ |y - f(x)| \le \delta, \\ \delta|y - f(x)| - \frac{1}{2}\delta^2 & otherwise. \end{cases} \quad (16)$$

All numeric variables in the input variables were standardized. The Input variable formats of the MLP,1-D CNN, and RICNN are presented in Tables 11–13, respectively.

In the case of MLP and 1-D CNN, we performed the experiments with and without dropout [78] for the dense layer. In RNN and RICNN, the dropout was activated only on the LSTM layer [79]. All dropout rates are set to 0.8.

5.3. Performance measure

RMSE and MAPE were used to evaluate the performance of the forecasting models. RMSE and MAPE equations are shown in Eqs. (17) and (18), respectively. y_t is the actual value of the t time,

$f(x)_t$ is the predicted value of the t time, and n is the number of observations.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{t=1}^{n} (f(x)_t - y_t)^2}{n}} \tag{17}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{f(x)_t - y_t}{y_t}\right| \times 100 \tag{18}$$

6. Results and discussion

6.1. Results of multi short-term load forecasting

The RMSEs and MAPEs of the three complexes with three learning days for each algorithm are presented in Tables 14–16. It is worth noting that the proposed RICNN exhibits the best performance in all cases except the Shihwa complex with learning periods of five days.

The average RMSEs and MAPEs of the four algorithms are presented in Table 17 and Fig. 8. It can be observed that the proposed RICNN yielded the best performance in terms of RMSE for the entire length of the training period in the three complexes. From a training period perspective, ANN and 1-D CNN resulted in the lowest RMSEs when the training period is short, i.e., three days, in all complexes, and their RMSEs increased in response to prolonged training periods. On the other hand, RNN and RICNN reported the best performance with three days of training in the two complexes

Table 14
Incheon cluster RMSE and MAPE results (numbers in parentheses denote the MAPE, the text in bold denote the best performances for each training days).

Algorithm	Dropout	3 Days	5 Days	7 Days
MLP-A	X	115.717 (7.079)	116.481 (7.237)	124.062 (7.596)
MLP-A	O	120.461 (7.696)	121.733 (7.823)	128.121 (8.395)
MLP-B	X	86.880 (5.812)	92.576 (6.251)	96.855 (6.531)
MLP-B	O	89.409 (5.906)	92.511 (6.084)	99.713 (6.670)
1-D CNN-A	X	102.583 (5.984)	106.477 (6.730)	106.100 (6.621)
1-D CNN-A	O	110.451 (7.036)	109.898 (7.108)	112.759 (6.857)
1-D CNN-B	X	79.579 (5.189)	83.258 (5.291)	85.565 (5.517)
1-D CNN-B	O	78.016 (5.071)	83.200 (5.251)	81.018 (5.303)
RNN-A	X	89.644 (5.096)	78.968 (4.879)	82.947 (4.873)
RNN-A	O	102.128 (5.138)	87.178 (5.501)	81.907 (5.149)
RNN-B	X	79.555 (4.817)	84.711 (5.447)	100.763 (6.121)
RNN-B	O	98.765 (4.854)	80.900 (4.883)	84.749 (5.229)
RICNN-A	X	89.340 (4.738)	82.075 (4.739)	79.826 (4.890)
RICNN-A	O	84.317 (4.807)	75.962 (4.787)	68.240 (4.370)
RICNN-B	X	**74.792 (4.389)**	**68.536 (4.140)**	71.107 (4.317)
RICNN-B	O	84.885 (4.554)	73.104 (4.695)	**65.808 (4.347)**

Table 15
Gwangju cluster RMSE and MAPE results (numbers in parentheses denote the MAPE, the text in bold denote the best performances for each training days).

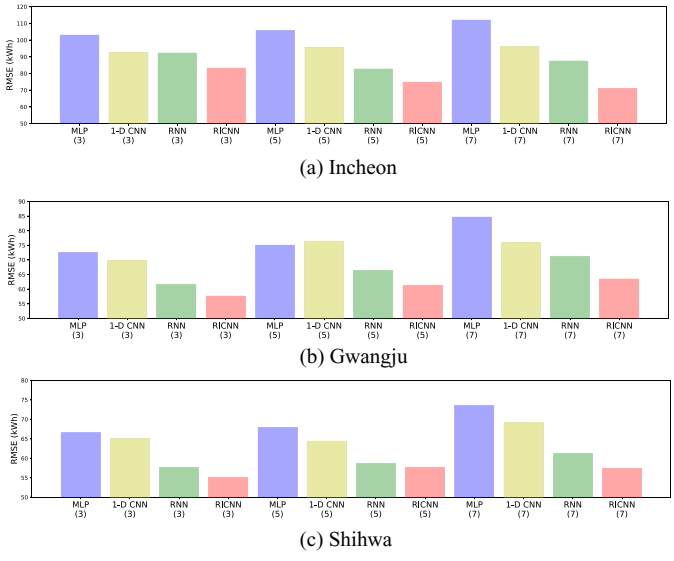
Algorithm	Dropout	3 Days	5 Days	7 Days
MLP-A	X	74.987 (6.794)	72.115 (6.573)	84.655 (7.807)
MLP-A	O	77.817 (6.742)	80.835 (7.118)	88.262 (7.817)
MLP-B	X	69.938 (6.847)	75.442 (7.092)	85.305 (8.084)
MLP-B	O	67.750 (6.375)	72.170 (6.871)	80.863 (7.914)
1-D CNN-A	X	71.364 (6.517)	74.728 (7.005)	75.282 (6.560)
1-D CNN-A	O	72.723 (6.430)	82.635 (7.125)	76.031 (6.570)
1-D CNN-B	X	66.816 (6.021)	74.010 (7.046)	80.219 (7.371)
1-D CNN-B	O	68.082 (6.379)	73.984 (6.376)	72.184 (6.749)
RNN-A	X	63.377 (5.132)	58.170 (5.281)	75.402 (5.787)
RNN-A	O	58.039 (4.751)	65.641 (5.417)	70.201 (5.608)
RNN-B	X	68.643 (5.365)	70.803 (5.704)	69.021 (5.634)
RNN-B	O	56.649 (4.820)	71.067 (5.482)	70.629 (5.729)
RICNN-A	X	56.171 (4.604)	62.590 (5.155)	77.785 (5.918)
RICNN-A	O	**51.958 (4.499)**	61.038 (4.958)	59.399 (4.908)
RICNN-B	X	66.052 (5.130)	63.586 (5.155)	61.726 (5.540)
RICNN-B	O	56.560 (4.642)	**57.985 (5.401)**	**55.366 (4.779)**

Table 16
Shihwa cluster RMSE and MAPE results (numbers in parentheses denote the MAPE, the text in bold denote the best performances for each training days).

Algorithm	Dropout	3 Days	5 Days	7 Days
MLP-A	X	66.002 (10.002)	68.795 (10.768)	75.425 (11.260)
MLP-A	O	71.938 (10.602)	70.660 (10.382)	75.138 (11.089)
MLP-B	X	62.995 (9.911)	65.763 (10.399)	70.229 (11.029)
MLP-B	O	65.420 (10.389)	66.619 (10.406)	73.772 (10.976)
1-D CNN-A	X	67.172 (10.310)	65.411 (10.356)	70.833 (11.052)
1-D CNN-A	O	66.344 (10.182)	66.412 (10.726)	70.607 (11.279)
1-D CNN-B	X	63.929 (10.252)	62.276 (10.279)	67.221 (10.936)
1-D CNN-B	O	62.539 (9.792)	63.623 (9.946)	67.765 (10.779)
RNN-A	X	54.888 (8.528)	60.583 (8.777)	59.326 (9.414)
RNN-A	O	56.251 (8.184)	**54.765 (8.409)**	61.971 (8.383)
RNN-B	X	60.478 (9.053)	61.071 (9.112)	60.142 (9.245)
RNN-B	O	58.685 (8.458)	58.073 (8.569)	63.605 (9.206)
RICNN-A	X	55.543 (8.367)	57.817 (9.068)	57.735 (8.996)
RICNN-A	O	**51.863 (8.086)**	55.936 (8.191)	**51.848 (7.832)**
RICNN-B	X	56.262 (8.435)	57.579 (9.161)	63.944 (9.567)
RICNN-B	O	56.906 (9.028)	58.877 (9.134)	56.375 (8.761)

Table 17
Total average RMSE and MAPE results (numbers in parentheses denote the MAPE, the text in bold denote the best performances for each training day).

Area	Algorithm	3 Days	5 Days	7 Days
Incheon	MLP	103.117 (6.623)	105.825 (6.849)	112.188 (7.298)
	1-D CNN	92.657 (5.820)	95.719 (6.095)	96.361 (6.074)
	RNN	92.523 (4.976)	82.939 (5.177)	87.592 (5.343)
	RICNN	**83.333 (4.622)**	**74.919 (4.590)**	**71.245 (4.481)**
Gwangju	MLP	72.623 (6.689)	75.141 (6.914)	84.771 (7.905)
	1-D CNN	69.746 (6.337)	76.339 (6.888)	75.929 (6.813)
	RNN	61.677 (5.017)	66.420 (5.471)	71.313 (5.689)
	RICNN	**57.685 (4.719)**	**61.300 (5.167)**	**63.569 (5.286)**
Shihwa	MLP	66.589 (10.226)	67.959 (10.489)	73.641 (11.088)
	1-D CNN	64.996 (10.134)	64.431 (10.327)	69.106 (11.012)
	RNN	57.576 (8.556)	58.623 (8.717)	61.261 (9.062)
	RICNN	**55.144 (8.479)**	**57.552 (8.889)**	**57.476 (8.789)**



(a) Incheon

(b) Gwangju

(c) Shihwa

Fig. 8. RMSE results for each algorithm and training days (numbers in parentheses indicate the training days).

(Gwangju and Shihwa), but the lowest RMSEs are achieved with seven days of training in the Incheon complex, which means that the best training periods are different in actual environments according to the characteristics of the complex and external features.

The average RMSEs and MAPEs of the 48 prediction points with a 30-min interval for each algorithm are presented in Tables 18–20, and they are summarized as shown in Fig. 9. In the case of

Table 18
Performance result by timestamp based on each algorithm (numbers in parentheses denote the MAPE, the text in bold denote the best performances for each time).

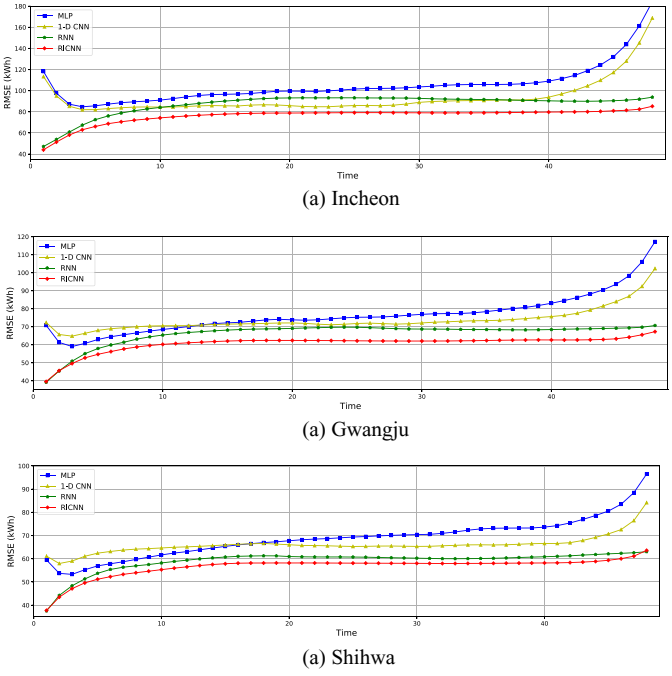
Time	Incheon				Gwangju				Shihwa			
	MLP	1-D CNN	RNN	RICNN	MLP	1-D CNN	RNN	RICNN	MLP	1-D CNN	RNN	RICNN
1	118.249	113.371	47.123	**43.973**	70.779	72.340	**39.158**	39.573	59.476	61.131	**37.536**	37.763
	(8.863)	(8.338)	(3.299)	**(2.993)**	(7.542)	(7.328)	**(3.681)**	(3.746)	(9.523)	(9.812)	**(5.297)**	(5.291)
2	97.623	95.039	53.835	**51.326**	61.293	65.702	**45.451**	45.614	53.710	57.967	44.228	**43.478**
	(7.171)	(6.849)	(3.642)	**(3.374)**	(6.468)	(6.501)	**(4.068)**	(4.121)	(8.678)	(9.326)	(6.183)	**(6.231)**
3	87.219	85.210	60.830	**58.047**	59.029	64.784	50.871	**49.576**	53.281	59.026	48.363	**47.137**
	(6.247)	(6.077)	(3.934)	**(3.690)**	(6.123)	(6.293)	(4.411)	**(4.363)**	(8.509)	(9.392)	(6.837)	**(6.900)**
4	84.743	81.988	67.383	**62.989**	60.757	66.368	55.054	**52.749**	55.274	61.077	51.325	**49.612**
	(5.972)	(5.793)	(4.197)	**(3.892)**	(6.214)	(6.354)	(4.648)	**(4.551)**	(8.684)	(9.606)	(7.308)	**(7.358)**
5	85.646	82.053	72.534	**66.106**	62.947	67.995	57.914	**54.695**	56.931	62.414	53.725	**51.179**
	(5.976)	(5.735)	(4.401)	**(4.021)**	(6.373)	(6.426)	(4.824)	**(4.660)**	(8.874)	(9.789)	(7.666)	**(7.679)**
6	87.277	83.067	76.093	**68.756**	64.470	68.915	59.883	**56.228**	57.848	63.149	55.400	**52.288**
	(6.054)	(5.748)	(4.536)	**(4.145)**	(6.467)	(6.434)	(4.940)	**(4.753)**	(9.023)	(9.921)	(7.937)	**(7.920)**
7	88.630	83.838	78.833	**70.561**	65.563	69.467	61.381	**57.672**	58.711	63.754	56.357	**53.285**
	(6.155)	(5.745)	(4.638)	**(4.240)**	(6.497)	(6.440)	(5.041)	**(4.836)**	(9.155)	(10.055)	(8.134)	**(8.132)**
8	89.478	84.452	80.935	**72.035**	66.596	69.999	63.118	**58.768**	59.716	64.157	56.955	**53.988**
	(6.214)	(5.739)	(4.728)	**(4.313)**	(6.557)	(6.457)	(5.153)	**(4.897)**	(9.305)	(10.155)	(8.279)	**(8.296)**
9	90.149	84.619	82.689	**73.170**	67.601	70.401	64.399	**59.595**	60.697	64.375	57.509	**54.638**
	(6.208)	(5.732)	(4.819)	**(4.375)**	(6.632)	(6.475)	(5.245)	**(4.956)**	(9.472)	(10.230)	(8.407)	**(8.416)**
10	91.049	84.442	84.142	**74.228**	68.489	70.550	65.392	**60.209**	61.664	64.654	58.199	**55.316**
	(6.205)	(5.728)	(4.904)	**(4.429)**	(6.707)	(6.498)	(5.325)	**(5.005)**	(9.646)	(10.294)	(8.536)	**(8.533)**
11	92.462	84.543	85.433	**75.164**	69.272	70.571	66.187	**60.675**	62.471	64.952	58.838	**55.961**
	(6.248)	(5.720)	(4.981)	**(4.481)**	(6.760)	(6.508)	(5.388)	**(5.039)**	(9.784)	(10.337)	(8.662)	**(8.639)**
12	94.112	85.016	86.697	**76.013**	70.079	70.600	66.814	**61.090**	63.077	65.142	59.396	**56.528**
	(6.359)	(5.750)	(5.055)	**(4.530)**	(6.831)	(6.514)	(5.434)	**(5.071)**	(9.893)	(10.368)	(8.770)	**(8.736)**
13	95.468	85.619	87.944	**76.712**	71.009	70.974	67.306	**61.455**	63.867	65.366	59.950	**57.093**
	(6.475)	(5.797)	(5.120)	**(4.571)**	(6.917)	(6.536)	(5.474)	**(5.096)**	(10.012)	(10.416)	(8.860)	**(8.824)**
14	96.315	85.922	89.142	**77.312**	71.759	71.271	67.750	**61.779**	64.626	65.661	60.382	**57.535**
	(6.536)	(5.812)	(5.183)	**(4.610)**	(6.985)	(6.567)	(5.504)	**(5.121)**	(10.125)	(10.476)	(8.929)	**(8.892)**
15	96.579	85.684	90.147	**77.820**	72.155	71.459	68.141	**62.059**	65.344	65.949	60.747	**57.860**
	(6.557)	(5.759)	(5.236)	**(4.641)**	(7.010)	(6.605)	(5.526)	**(5.143)**	(10.248)	(10.527)	(8.985)	**(8.946)**
16	96.831	85.522	90.972	**78.218**	72.481	71.570	68.449	**62.254**	65.983	66.243	61.056	**58.092**
	(6.574)	(5.707)	(5.281)	**(4.659)**	(7.028)	(6.644)	(5.549)	**(5.160)**	(10.359)	(10.592)	(9.029)	**(8.981)**

Table 19
Performance result by timestamp based on each algorithm (numbers in parentheses denote the MAPE, the text in bold denote the best performances for each time).

Time	Incheon				Gwangju				Shihwa			
	MLP	1-D CNN	RNN	RICNN	MLP	1-D CNN	RNN	RICNN	MLP	1-D CNN	RNN	RICNN
17	97.366	86.353	91.793	**78.519**	73.170	71.698	68.670	**62.366**	66.493	66.466	61.191	**58.169**
	(6.631)	(5.758)	(5.332)	**(4.674)**	(7.089)	(6.645)	(5.567)	**(5.171)**	(10.440)	(10.683)	(9.061)	**(9.005)**
18	98.407	86.681	92.505	**78.718**	73.809	71.884	68.803	**62.410**	66.962	66.614	61.276	**58.175**
	(6.727)	(5.804)	(5.382)	**(4.689)**	(7.114)	(6.638)	(5.584)	**(5.181)**	(10.529)	(10.762)	(9.089)	**(9.025)**
19	99.466	86.470	92.978	**78.810**	74.129	72.079	68.916	**62.401**	67.314	66.323	61.255	**58.200**
	(6.801)	(5.839)	(5.416)	**(4.698)**	(7.121)	(6.661)	(5.601)	**(5.188)**	(10.602)	(10.767)	(9.103)	**(9.044)**
20	99.641	85.836	93.123	**78.820**	73.833	72.123	69.076	**62.403**	67.766	66.040	60.964	**58.185**
	(6.818)	(5.832)	(5.435)	**(4.702)**	(7.085)	(6.701)	(5.617)	**(5.195)**	(10.686)	(10.739)	(9.107)	**(9.061)**
21	99.547	85.056	93.234	**78.883**	73.618	71.855	69.308	**62.372**	68.200	65.757	60.845	**58.200**
	(6.828)	(5.774)	(5.452)	**(4.706)**	(7.050)	(6.706)	(5.635)	**(5.193)**	(10.759)	(10.689)	(9.113)	**(9.072)**
22	99.294	84.796	93.194	**78.968**	73.848	71.411	69.483	**62.342**	68.481	65.692	60.790	**58.186**
	(6.791)	(5.747)	(5.466)	**(4.705)**	(7.061)	(6.664)	(5.646)	**(5.188)**	(10.789)	(10.661)	(9.127)	**(9.074)**
23	99.825	84.797	93.207	**79.048**	74.395	71.101	69.665	**62.303**	68.692	65.591	60.776	**58.168**
	(6.796)	(5.742)	(5.483)	**(4.709)**	(7.089)	(6.602)	(5.650)	**(5.187)**	(10.800)	(10.653)	(9.142)	**(9.074)**
24	100.590	85.470	93.246	**79.139**	74.933	71.382	69.752	**62.251**	69.029	65.423	60.782	**58.138**
	(6.839)	(5.796)	(5.506)	**(4.718)**	(7.147)	(6.603)	(5.645)	**(5.186)**	(10.841)	(10.638)	(9.151)	**(9.069)**
25	101.559	85.920	93.239	**79.179**	75.270	71.810	69.707	**62.184**	69.364	65.299	60.744	**58.118**
	(6.957)	(5.817)	(5.520)	**(4.722)**	(7.188)	(6.653)	(5.641)	**(5.181)**	(10.88)	(10.636)	(9.154)	**(9.064)**
26	102.056	85.954	93.145	**79.208**	75.316	71.992	69.486	**62.141**	69.584	65.350	60.684	**58.105**
	(7.032)	(5.814)	(5.523)	**(4.723)**	(7.218)	(6.691)	(5.624)	**(5.179)**	(10.916)	(10.631)	(9.152)	**(9.063)**
27	102.158	85.829	93.072	**79.203**	75.438	71.741	69.201	**62.099**	69.855	65.469	60.563	**58.084**
	(7.035)	(5.763)	(5.521)	**(4.716)**	(7.243)	(6.691)	(5.603)	**(5.176)**	(10.985)	(10.653)	(9.151)	**(9.062)**
28	102.310	86.267	93.042	**79.157**	75.872	71.460	68.948	**62.078**	70.083	65.500	60.450	**58.062**
	(7.019)	(5.771)	(5.513)	**(4.710)**	(7.280)	(6.680)	(5.587)	**(5.175)**	(11.031)	(10.665)	(9.148)	**(9.06)**
29	102.742	87.356	92.987	**79.098**	76.424	71.658	68.763	**62.053**	70.260	65.400	60.354	**58.052**
	(7.035)	(5.852)	(5.493)	**(4.700)**	(7.324)	(6.700)	(5.569)	**(5.171)**	(11.055)	(10.682)	(9.147)	**(9.059)**
30	103.542	88.943	92.736	**79.053**	76.889	72.117	68.708	**62.063**	70.413	65.326	60.265	**58.032**
	(7.111)	(5.959)	(5.466)	**(4.691)**	(7.345)	(6.715)	(5.558)	**(5.167)**	(11.076)	(10.676)	(9.145)	**(9.057)**
31	104.215	89.737	92.336	**79.001**	77.217	72.533	68.687	**62.072**	70.529	65.370	60.151	**57.996**
	(7.177)	(5.993)	(5.440)	**(4.691)**	(7.356)	(6.731)	(5.547)	**(5.157)**	(11.097)	(10.669)	(9.141)	**(9.056)**
32	105.122	90.117	92.052	**78.966**	77.311	72.747	68.628	**62.097**	70.951	65.599	60.041	**57.963**
	(7.218)	(5.982)	(5.422)	**(4.692)**	(7.327)	(6.714)	(5.536)	**(5.149)**	(11.152)	(10.693)	(9.143)	**(9.057)**

Table 20
Performance result by timestamp based on each algorithm (numbers in parentheses denote the MAPE, the text in bold denote the best performances for each time).

Time	Incheon				Gwangju				Shihwa			
	MLP	1-D CNN	RNN	RICNN	MLP	1-D CNN	RNN	RICNN	MLP	1-D CNN	RNN	RICNN
33	105.627	90.492	91.866	**78.984**	77.384	73.087	68.508	**62.183**	71.549	65.784	60.021	**57.972**
	(7.227)	(5.959)	(5.410)	**(4.690)**	(7.300)	(6.720)	(5.528)	**(5.143)**	(11.219)	(10.704)	(9.147)	**(9.058)**
34	105.835	90.619	91.745	**78.992**	77.718	73.315	68.429	**62.249**	72.305	65.989	60.058	**58.002**
	(7.223)	(5.954)	(5.399)	**(4.689)**	(7.302)	(6.729)	(5.525)	**(5.138)**	(11.310)	(10.711)	(9.153)	**(9.056)**
35	105.858	90.773	91.616	**79.058**	78.417	73.388	68.423	**62.361**	72.869	66.023	60.119	**58.018**
	(7.208)	(5.974)	(5.391)	**(4.688)**	(7.376)	(6.73)	(5.528)	**(5.138)**	(11.377)	(10.700)	(9.161)	**(9.053)**
36	105.900	91.106	91.447	**79.178**	79.249	73.596	68.350	**62.487**	73.147	65.957	60.206	**58.042**
	(7.201)	(5.997)	(5.382)	**(4.687)**	(7.438)	(6.724)	(5.527)	**(5.141)**	(11.409)	(10.648)	(9.163)	**(9.050)**
37	106.022	91.191	91.138	**79.285**	80.066	73.995	68.286	**62.581**	73.212	66.030	60.361	**58.089**
	(7.168)	(5.962)	(5.369)	**(4.690)**	(7.502)	(6.734)	(5.519)	**(5.144)**	(11.417)	(10.618)	(9.172)	**(9.054)**
38	106.526	91.160	90.869	**79.443**	80.736	74.508	68.255	**62.645**	73.241	66.257	60.544	**58.134**
	(7.163)	(5.901)	(5.361)	**(4.696)**	(7.522)	(6.756)	(5.523)	**(5.144)**	(11.409)	(10.605)	(9.181)	**(9.056)**
39	107.480	91.947	90.621	**79.598**	81.782	75.125	68.319	**62.689**	73.258	66.495	60.714	**58.168**
	(7.200)	(5.879)	(5.355)	**(4.706)**	(7.554)	(6.772)	(5.524)	**(5.144)**	(11.397)	(10.615)	(9.191)	**(9.053)**
40	109.055	93.847	90.401	**79.737**	83.047	75.663	68.426	**62.664**	73.645	66.602	60.868	**58.191**
	(7.240)	(5.924)	(5.348)	**(4.721)**	(7.571)	(6.759)	(5.526)	**(5.143)**	(11.420)	(10.618)	(9.200)	**(9.050)**
41	111.384	96.921	90.156	**79.818**	84.541	76.342	68.613	**62.647**	74.281	66.594	61.062	**58.246**
	(7.280)	(6.034)	(5.343)	**(4.739)**	(7.615)	(6.737)	(5.534)	**(5.143)**	(11.447)	(10.587)	(9.213)	**(9.052)**
42	114.459	100.285	90.038	**79.922**	86.207	77.496	68.756	**62.660**	75.433	66.939	61.270	**58.363**
	(7.280)	(6.099)	(5.348)	**(4.764)**	(7.663)	(6.761)	(5.541)	**(5.148)**	(11.498)	(10.595)	(9.226)	**(9.053)**
43	118.950	104.582	89.967	**80.089**	88.193	79.297	68.881	**62.762**	76.944	67.891	61.597	**58.574**
	(7.313)	(6.164)	(5.361)	**(4.794)**	(7.709)	(6.840)	(5.551)	**(5.166)**	(11.550)	(10.662)	(9.238)	**(9.060)**
44	124.336	109.895	90.138	**80.385**	90.483	81.568	69.016	**62.924**	78.629	69.228	61.835	**58.858**
	(7.270)	(6.177)	(5.391)	**(4.809)**	(7.731)	(6.924)	(5.561)	**(5.200)**	(11.581)	(10.763)	(9.244)	**(9.072)**
45	132.108	117.279	90.480	**80.926**	93.616	83.971	69.182	**63.305**	80.554	70.706	62.123	**59.351**
	(7.213)	(6.139)	(5.424)	**(4.839)**	(7.717)	(6.931)	(5.577)	**(5.281)**	(11.565)	(10.804)	(9.261)	**(9.105)**
46	143.837	128.065	91.021	**81.441**	98.352	86.928	69.348	**64.198**	83.493	72.577	62.350	**60.039**
	(7.316)	(6.191)	(5.463)	**(4.996)**	(7.702)	(6.845)	(5.606)	**(5.312)**	(11.557)	(10.753)	(9.277)	**(9.224)**
47	161.461	145.368	91.936	**82.434**	106.09	92.435	69.725	**65.553**	88.487	76.468	62.594	**61.128**
	(7.765)	(6.639)	(5.543)	**(5.087)**	(7.883)	(6.901)	(5.664)	**(5.416)**	(11.684)	(10.803)	(9.299)	**(9.325)**
48	184.278	168.833	93.839	**85.234**	117.232	102.327	70.676	**67.285**	96.533	84.197	**63.036**	63.615
	(8.625)	(7.564)	(5.738)	**(5.286)**	(8.385)	(7.358)	(5.781)	**(5.590)**	(12.082)	(11.175)	**(9.333)**	(9.507)



(a) Incheon

(a) Gwangju

(a) Shihwa

Fig. 9. Average RMSE results of each time point and algorithms.

Table 21
Time consumption during 1000 iterations for training each algorithms (Unit: Second, Batch size=256, Measure by RTX 2080-Ti GPU).

Algorithm	3 Days	5 Days	7 Days
MLP-A	19.47	27.08	33.49
1-D CNN-A	27.08	34.86	40.77
RNN-A	223.58	326.44	421.19
RICNN-A	233.28	338.81	425.52
MLP-B	21.09	27.62	33.87
1-D CNN-B	30.96	41.15	49.56
RNN-B	236.42	341.14	432.59
RICNN-B	244.30	364.45	456.58

Table 22
Time consumption during inference batch data for each algorithms (Unit: Millisecond, Batch size=256, Measure by RTX 2080-Ti GPU).

Algorithm	3 Days	5 Days	7 Days
MLP-A	5.12	7.68	10.24
1-D CNN-A	6.12	11.38	13.42
RNN-A	15.36	23.04	30.72
RICNN-A	17.92	25.60	33.28
MLP-B	6.08	8.14	11.19
1-D CNN-B	7.68	12.62	15.36
RNN-B	28.16	40.96	53.76
RICNN-B	30.72	48.64	58.88

the Incheon complex, the proposed RICNN always results in the best performance for all sequences. For the Gwangju complex, RNN yielded the best performance for the first two points (30-min, 1-h), but the proposed RICNN demonstrated the lowest RMSEs for the other points. For the Shihwa complex, RICNN resulted in the lowest RMSEs for all prediction points except for the last one (48th point).

According to the Fig. 9, it is observed that the algorithms exhibit two types of the RMSE trends: an U-shape (MLP and 1-D CNN) or an S-shape (RNN and RICNN), both of which have a wide plateau in the middle. The load forecasts based on MLP or 1-D CNN begin with a relatively large RMSE but stabilizes after the third or fourth prediction points. The RMSE then becomes stable or marginally increases. However, it rapidly increases from a certain point to the end. The load forecasts based on RNN or RICNN,

on the other hand, begin with a very low RMSE and increase to a certain level, after which the RMSE becomes stable. In contrast to the MLP and 1-D CNN, RNN and RICNN maintain favorable prediction ability until the end of the prediction point. When comparing the RNN and RICNN, the forecast performance at the beginning is similar to each other, but the RICNN eventually yields consistently lower RMSEs than that of the RNN.

At the time of prediction, although RNN uses only the current hidden state vector, RICNN uses 1-D convolution inception modules associated with current and nearby neighbor hidden state vectors. Based on our experiment, we can conclude that the proposed RICNN has the ability to adjust formerly made unfavorable hidden state vectors, which in turn, contributes to lower electric load forecast errors than that of the RNN-based forecast model.

7. Conclusion

This study proposed a new multi short-term electrical load forecasting model, termed RICNN, which combines RNN and a 1-D inception module. The RNN model preserves past information while the 1-D convolution inception module helps calibrate the prediction time and the hidden state vector values calculated from the nearby time steps. To verify the proposed RICNN model, we used actual smart grid data collected from three industrial distribution complexes in South Korea. 21 input variables are used each day during three different training periods, i.e., three, five, and seven days are tested to predict the electrical load of the following day (48 time points with an interval of 30 min). Based on the experimental results, we observed that the proposed RICNN always yielded better forecasting performance than MLP, 1-D CNN, and RNN in terms of RMSE. In particular, the proposed RICNN demonstrates robustness to the prediction time points; its RMSE did not increase significantly while those of other benchmark models did.

Despite favorable experimental results, the current study has several limitations, providing us with future research directions. First, Tables 21 and 22 show that RNN-based load forecasting algorithms consumed more training and inference time than MLP and CNN-based algorithms. Therefore, it is necessary to improve the training and inference efficiencies of the proposed model while preserving the good forecasting performances. Second, this study executed a 30-min VSTLF. Therefore, the RICNN model should be verified under additional varied prediction time intervals such as 15 min, an hour, or a day.

Acknowledgments

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B03930729) and Korea Electric Power Corporation (Grant number: R18XA05).

References

[1] W.W. Kim, J.S. Shin, J.O. Kim, Operation strategy of multi-energy storage system for ancillary services, IEEE Trans. Power Syst. 32 (6) (2017) 4409–4417, doi:10.1109/TPWRS.2017.2665669.
[2] T. Niet, B. Lyseng, J. English, V. Keller, K. Palmer-Wilson, I. Moazzen, et al., Hedging the risk of increased emissions in long term energy planning, Energy Strat. Rev. 16 (2017) 1–12, doi:10.1016/j.esr.2017.02.001.
[3] C. Cui, T. Wu, M. Hu, J.D. Weir, X. Li, Short-term building energy model recommendation system: a meta-learning approach, Appl. Energy 172 (2016) 251–263, doi:10.1016/j.apenergy.2016.03.112.
[4] M.W. Ahmad, M. Mourshed, Y. Rezgui, Trees vs Neurons: comparison between random forest and ANN for high-resolution prediction of building energy consumption, Energy Build. 147 (2017) 77–89, doi:10.1016/j.enbuild.2017.04.038.
[5] S.K. Lee, G. Mogi, Relative efficiency of energy technologies in the Korean mid-term strategic energy technology development plan, Renew. Sustain. Energy Rev. 91 (2018) 472–482, doi:10.1016/j.rser.2018.03.031.
[6] J. Jeong, T. Hong, C. Ji, J. Kim, M. Lee, K. Jeong, et al., Improvements of the operational rating system for existing residential buildings, Appl. Energy 193 (2017) 112–124, doi:10.1016/j.apenergy.2017.02.036.
[7] M.H. Chung, E.K. Rhee, Potential opportunities for energy conservation in existing buildings on university campus: a field survey in Korea, Energy Build. 78 (2014) 176–182, doi:10.1016/j.enbuild.2014.04.018.
[8] L. Bolívar Jaramillo, A. Weidlich, Optimal microgrid scheduling with peak load reduction involving an electrolyzer and flexible loads, Appl. Energy 169 (2016) 857–865, doi:10.1016/j.apenergy.2016.02.096.
[9] C. Fan, F. Xiao, Y. Zhao, A short-term building cooling load prediction method using deep learning algorithms, Appl. Energy 195 (2017) 222–233, doi:10.1016/j.apenergy.2017.03.064.
[10] Z. Zhang, J. Wang, X. Wang, An improved charging/discharging strategy of lithium batteries considering depreciation cost in day-ahead microgrid scheduling, Energy Convers. Manag. 105 (2015) 675–684, doi:10.1016/j.enconman.2015.07.079.
[11] J. Li, X. Wang, Z. Zhang, S. Le Blond, Q. Yang, M. Zhang, et al., Analysis of a new design of the hybrid energy storage system used in the residential m-CHP systems, Appl. Energy 187 (2017) 169–179, doi:10.1016/j.apenergy.2016.11.058.
[12] M. Short, T. Crosbie, M. Dawood, N. Dawood, Load forecasting and dispatch optimisation for decentralised co-generation plant with dual energy storage, Appl. Energy 186 (2017) 304–320, doi:10.1016/j.apenergy.2016.04.052.
[13] S. Parhizi, A. Khodaei, M. Shahidehpour, Market-based versus price-based microgrid optimal scheduling, IEEE Trans. Smart Grid 9 (2) (2018) 615–623, doi:10.1109/TSG.2016.2558517.
[14] T. Hong, S. Fan, Probabilistic electric load forecasting: a tutorial review, Int. J. Forecast. 32 (3) (2016) 914–938, doi:10.1016/j.ijforecast.2015.11.011.
[15] A. Dedinec, S. Filiposka, A. Dedinec, L. Kocarev, Deep belief network based electricity load forecasting: an analysis of Macedonian case, Energy 115 (2016) 1688–1700, doi:10.1016/j.energy.2016.07.090.
[16] E. Mocanu, P.H. Nguyen, M. Gibescu, W.L. Kling, Deep learning for estimating building energy consumption, Sustain. Energy Grids Netw. 6 (2016) 91–99, doi:10.1016/j.segan.2016.02.005.
[17] B. Zhang, J.-L. Wu, P.-C. Chang, A multiple time series-based recurrent neural network for short-term load forecasting, Soft Comput. 22 (12) (2017) 4099–4112, doi:10.1007/s00500-017-2624-5.
[18] M. Abdel-Nasser, K. Mahmoud, Accurate photovoltaic power forecasting models using deep LSTM-RNN, Neural Comput. Appl. (2017) 1–14, doi:10.1007/s00521-017-3225-z.
[19] D.L. Marino, K. Amarasinghe, M. Manic, Building energy load forecasting using Deep Neural Networks, in: IECON 2016 - 42nd Annu Conf IEEE Ind Electron Soc, 2016, pp. 7046–7051, doi:10.1109/IECON.2016.7793413.
[20] H. Shi, M. Xu, R. Li, Deep learning for household load forecasting – a novel pooling deep RNN, IEEE Trans. Smart Grid 3053 (5) (2017) 1–1, doi:10.1109/TSG.2017.2686012.
[21] W. Kong, Z.Y. Dong, Y. Jia, D.J. Hill, Y. Xu, Y. Zhang, Short-term residential load forecasting based on LSTM recurrent neural network, IEEE Trans. Smart Grid 3053 (1) (2017) 1–11, doi:10.1109/TSG.2017.2753802.
[22] B. Yildiz, J.I. Bilbao, A.B. Sproul, A review and analysis of regression and machine learning models on commercial building electricity load forecasting, Renew. Sustain. Energy Rev. 73 (2017) 1104–1122, doi:10.1016/j.rser.2017.02.023.
[23] M.Q. Raza, A. Khosravi, A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings, Renew. Sustain. Energy Rev. 50 (2015) 1352–1372, doi:10.1016/j.rser.2015.04.065.
[24] L. Hernandez, C. Baladron, J.M. Aguiar, B. Carro, a.J. Sanchez-Esguevillas, J. Lloret, et al., A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1460–1495, doi:10.1109/SURV.2014.032014.00094.
[25] A. Almalaq, G. Edwards, A review of deep learning methods applied on load forecasting, in: 2017 16th IEEE Int Conf Mach Learn Appl, 2017, pp. 511–516, doi:10.1109/ICMLA.2017.0-110.
[26] A. Vaghefi, M.A. Jafari, E. Bisse, Y. Lu, J. Brouwer, Modeling and forecasting of cooling and electricity load demand, Appl. Energy 136 (2015) 186–196, doi:10.1016/j.apenergy.2014.09.004.
[27] G. Dudek, Pattern-based local linear regression models for short-term load forecasting, Electr. Power Syst. Res. 130 (2016) 139–147, doi:10.1016/j.epsr.2015.09.001.
[28] A.K. Fard, M.-R. Akbari-Zadeh, A hybrid method based on wavelet, ANN and ARIMA model for short-term load forecasting, J. Exp. Theor. Artif. Intell. 26 (2) (2014) 167–182, doi:10.1080/0952813X.2013.813976.
[29] G. Sudheer, A. Suseelatha, Short term load forecasting using wavelet transform combined with Holt-Winters and weighted nearest neighbor models, Int. J. Electr. Power Energy Syst. 64 (2015) 340–346, doi:10.1016/j.ijepes.2014.07.043.
[30] X. Ke, A. Jiang, N. Lu, Load profile analysis and short-term building load forecast for a university campus, in: 2016 IEEE Power Energy Soc Gen Meet, 2016, pp. 1–5, doi:10.1109/PESGM.2016.7742034.
[31] A. Gerossier, R. Girard, G. Kariniotakis, A. Michiorri, A. Gerossier, R. Girard, et al., Probabilistic day-ahead forecasting of household electricity demand, in: 24th Int Conf Electr Distrib, 2017, Jun 2017, p. 0625.
[32] Y. Chen, H. Tan, Short-term prediction of electric demand in building sector via hybrid support vector regression, Appl. Energy 204 (2017) 1363–1374, doi:10.1016/j.apenergy.2017.03.070.
[33] R.K. Jain, K.M. Smith, P.J. Culligan, J.E. Taylor, Forecasting energy consumption of multi-family residential buildings using support vector regression: investigating the impact of temporal and spatial monitoring granularity on

performance accuracy, Appl. Energy 123 (2014) 168–178, doi:10.1016/j.apenergy.2014.02.057.
[34] K.P. Amber, M.W. Aslam, S.K. Hussain, Electricity consumption forecasting models for administration buildings of the UK higher education sector, Energy Build. 90 (2015) 127–136, doi:10.1016/j.enbuild.2015.01.008.
[35] K. Grolinger, A. L'Heureux, M.A.M. Capretz, L. Seewald, Energy forecasting for event venues: big data and prediction accuracy, Energy Build. 112 (2016) 222–233, doi:10.1016/j.enbuild.2015.12.010.
[36] S. Jurado, À. Nebot, F. Mugica, N. Avellana, Hybrid methodologies for electricity load forecasting: entropy-based feature selection with machine learning and soft computing techniques, Energy 86 (2015) 276–291, doi:10.1016/j.energy.2015.04.039.
[37] J. Moon, K. Kim, Y. Kim, E. Hwang, 2018 IEEE International Conference on Big Data and Smart Computing A Short-Term Electric Load Forecasting Scheme Using 2-Stage Predictive Analytics, 2018, doi:10.1109/BigComp.2018.00040.
[38] Hong T., Maciejowska K., Nowotarski J.. Probabilistic load forecasting via quantile regression averaging of independent expert forecasts 2014;8:1–3. doi:10.1109/TSG.2015.2437877.
[39] K. Li, C. Hu, G. Liu, W. Xue, Building's electricity consumption prediction using optimized artificial neural networks and principal component analysis, Energy Build. 108 (2015) 106–113, doi:10.1016/j.enbuild.2015.09.002.
[40] K. Li, H. Su, J. Chu, Forecasting building energy consumption using neural networks and hybrid neuro-fuzzy system: a comparative study, Energy Build. 43 (10) (2011) 2893–2899, doi:10.1016/j.enbuild.2011.07.010.
[41] A. Bagnasco, F. Fresi, M. Saviozzi, F. Silvestro, A. Vinci, Electrical consumption forecasting in hospital facilities: an application case, Energy Build. 103 (2015) 261–270, doi:10.1016/j.enbuild.2015.05.056.
[42] H. Chitsaz, H. Shaker, H. Zareipour, D. Wood, N. Amjady, Short-term electricity load forecasting of buildings in microgrids, Energy Build. 99 (2015) 50–60, doi:10.1016/j.enbuild.2015.04.011.
[43] R. Hu, S. Wen, Z. Zeng, T. Huang, A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm, Neurocomputing 221 (2017) 24–31, doi:10.1016/j.neucom.2016.09.027.
[44] N. Zeng, H. Zhang, W. Liu, J. Liang, F.E. Alsaadi, A switching delayed PSO optimized extreme learning machine for short-term load forecasting, Neurocomputing 240 (2017) 175–182, doi:10.1016/j.neucom.2017.01.090.
[45] S.S. Reddy, Bat algorithm-based back propagation approach for short-term load forecasting considering weather factors, Electr. Eng. 100 (3) (2017) 1–7, doi:10.1007/s00202-017-0587-2.
[46] M. Mordjaoui, S. Haddad, A. Medoued, A. Laouafi, Electric load forecasting by using dynamic neural network, Int. J. Hydrogen Energy 42 (28) (2017) 17655–17663, doi:10.1016/j.ijhydene.2017.03.101.
[47] S. Ryu, J. Noh, H. Kim, Deep neural network based demand side short term load forecasting, Energies 10 (2017) 1–20, doi:10.3390/en10010003.
[48] P.-H. Kuo, C.-J. Huang, A high precision artificial neural networks model for short-term energy load forecasting, Energies 11 (1) (2018) 213, doi:10.3390/en11010213.
[49] J. Pascual, J. Barricarte, P. Sanchis, L. Marroyo, Energy management strategy for a renewable-based residential microgrid with generation and demand forecasting, Appl. Energy 158 (2015) 12–25, doi:10.1016/j.apenergy.2015.08.040.
[50] C. Sandels, J. Widén, L. Nordström, E. Andersson, Day-ahead predictions of electricity consumption in a Swedish office building from weather, occupancy, and temporal data, Energy Build. 108 (2015) 279–290, doi:10.1016/j.enbuild.2015.08.052.
[51] J. Moon, J. Park, E. Hwang, S. Jun, Forecasting power consumption for higher educational institutions based on machine learning, J. Supercomput. 74 (8) (2017) 3778–3800, doi:10.1007/s11227-017-2022-x.
[52] W. Lee, J. Jung, M. Lee, Development of 24-hour optimal scheduling algorithm for energy storage system using load forecasting and renewable energy forecasting, in: 2017 IEEE Power Energy Soc Gen Meet, 2017, pp. 1–5, doi:10.1109/PESGM.2017.8273907.
[53] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 60 (6) (2012) 1–9. http://dx.doi.org/10.1016/j.protcy.2014.09.007.
[54] Simonyan K., Zisserman A.. Very deep convolutional networks for large-scale image recognition 2014:1–14. doi:10.1016/j.infsof.2008.09.005.
[55] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., et al. Going deeper with convolutions 2014:1–9. doi:10.1109/CVPR.2015.7298594.
[56] S. Wu, S. Zhong, Y. Liu, Deep residual learning for image steganalysis, Multimed. Tools Appl. 77 (9) (2017) 1–17, doi:10.1007/s11042-017-4440-4.
[57] G. Huang, Z. Liu, L.v.d Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: 2017 IEEE Conf Comput Vis Pattern Recognit, 2017, pp. 2261–2269, doi:10.1109/CVPR.2017.243.
[58] Kim Y.. Convolutional neural networks for sentence classification 2014:1746–51. doi:10.3115/v1/D14-1181.
[59] A.S. Ahmad, M.Y. Hassan, M.P. Abdullah, H.A. Rahman, F. Hussin, H. Abdullah, et al., A review on applications of ANN and SVM for building electrical energy consumption forecasting, Renew. Sustain. Energy Rev. 33 (2014) 102–109, doi:10.1016/j.rser.2014.01.069.
[60] H.-z. Wang, G.-q. Li, G.-b. Wang, J.-c. Peng, H. Jiang, Y.-t. Liu, Deep learning based ensemble approach for probabilistic wind power forecasting, Appl. Energy 188 (2017) 56–70, doi:10.1016/j.apenergy.2016.11.111.
[61] A. Graves, N. Jaitly, Towards end-to-end speech recognition with recurrent neural networks, in: JMLR Workshop Conf Proc, 32, 2014, pp. 1764–1772, doi:10.1145/1143844.1143891.
[62] A. Graves, A.-R. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: 2013 IEEE Int Conf Acoust Speech Signal Process, 2013, pp. 6645–6649, doi:10.1109/ICASSP.2013.6638947.
[63] Zhang X., LeCun Y.. Which encoding is the best for text classification in Chinese, English, Japanese and Korean?. arXiv preprint arXiv:1708.02657. 2017.
[64] Liu P., Qiu X., Huang X.. Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101. 2016.
[65] Kim J., Kang P.. Recurrent neural network-based user authentication for freely typed keystroke data. arXiv preprint arXiv:1806.06190. 2018.
[66] S. Hochreiter, J.J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1–32, doi:10.1162/neco.1997.9.8.1735.
[67] A. Dosovitskiy, J.T. Springenberg, T. Brox, An empirical exploration of recurrent network architectures, in: Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit, 2015, pp. 1538–1546, doi:10.1109/CVPR.2015.7298761. 07–12–June.
[68] K. Greff, R.K. Srivastava, J. Koutnik, B.R. Steunebrink, J. Schmidhuber, LSTM: a search space odyssey, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2017) 2222–2232, doi:10.1109/TNNLS.2016.2582924.
[69] P. Pinheiro, R. Collobert, Recurrent convolutional neural networks for scene labeling, in: Proc 31st Int Conf Mach Learn, 32, 2014, pp. 82–90.
[70] J. Donahue, L.A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, et al., Long-term recurrent convolutional networks for visual recognition and description, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017) 677–691, doi:10.1109/TPAMI.2016.2599174.
[71] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Twenty-Ninth AAAI Conf Artif Intell, 2015, pp. 2267–2273.
[72] K. Choi, G. Fazekas, M. Sandler, K. Cho, Convolutional recurrent neural networks for music classification, in: 2017 IEEE Int Conf Acoust Speech Signal Process, 2017, pp. 2392–2396, doi:10.1109/ICASSP.2017.7952585.
[73] Emre Cakir E., Adavanne S., Parascandolo G., Drossos K., Virtanen T.. Convolutional recurrent neural networks for bird audio detection. 2017 25th European Signal Processing Conference (EUSIPCO). doi:10.23919/EUSIPCO.2017.8081508.
[74] Ioffe S., Szegedy C.. Batch normalization: accelerating deep network training by reducing internal covariate shift 2015. doi:10.1007/s13398-014-0173-7.2.
[75] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: PMLR, 9, 2010, pp. 249–256. doi:10.1.1.207.2059.
[76] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural Networks Mach. Learn. (2012).
[77] J. Niu, J. Chen, Y. Xu, Twin support vector regression with Huber loss, J. Intell. Fuzzy Syst. 32 (2017) 4247–4258, doi:10.3233/JIFS-16629.
[78] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958, doi:10.1214/12-AOS1000.
[79] Zaremba W., Sutskever I., Vinyals O.. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329. 2014.